



UNITED STATES PATENT AND TRADEMARK OFFICE

AS.

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/903,100	07/11/2001	Ramesh Subramanian	GSH 08-885923	1887
27667	7590	01/26/2005	EXAMINER	
HAYES, SOLOWAY P.C. 130 W. CUSHING STREET TUCSON, AZ 85701			CAO, DIEM K	
			ART UNIT	PAPER NUMBER
			2126	

DATE MAILED: 01/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Applicant No.	Applicant(s)	
	09/903,100	SUBRAMANIAN ET AL.	
	Examiner Diem Cao	Art Unit 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 29 December 2004.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-34 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-34 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. Claims 1-34 are pending. This action is in response to the amendment filed 12/19/2004. Applicant has amended claims 1, 3, 7, 8, 13-15, 21, 24-27, 33 and 34. added claim 24.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The specification is objected to under 35 U.S.C. 112, first paragraph, as failing to adequately teach the claimed limitation “C++ based application” as recited in claims 1-34.

In the application as filed, there does not appear to be any detailed descriptions or disclosure of C++ based application capable of handling C++ exceptions, nor a thread executing a C++ based application. At best, applicant discloses window applications (page 2) which produces C-style exceptions (page 14). Applicant fails to disclose “C++ based application” capable of handling C++ exceptions in the specification as filed.

Claims 1-34 are rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Applicant reciteS “C++ based application” in claims 1-34. There does not appear to be a written description of the claimed limitation in the application as filed, for the reasons set forth in the objection to the specification.

4. Claims 1, 3, 4, 7, 13, 14, 21, 23, 24, 26, 33 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kannan et al. (U.S. 5,815,702) in view of Bak et al. (U.S. 6,415,381 B1).

As to claim 1, Kannan teaches receiving an exception caused due to a runtime fault in a thread executing the application (The current instruction 302 ... a general protection fault; col. 6, line 66 - col. 7, line 2), dispatching the exception to an exception handler (The operating system . . . to the chain of exception handlers; col. 7, lines 5-8), trapping the exception before the exception reaches the exception handler when the exception handler is a top level exception handler which terminates the application (The exception handler 115 ... terminate the application; col. 4, lines 44-47 and col. 1, line 48 - col. 2, line 11), and continuing execution of the application (the application to continue executing; col. 4, lines 58-64 and the exception handler ... in the proper manner; col. 7, lines 29-33).

Kannan does not teach translating the trapped exception into an exception that the application is capable of handling, nor the application is a C++ based application and the exception is a C++ exception.

Bak teaches translating the trapped exception into an exception that the application is capable of handling, including translating the trapped exception into a C++ exception (Java exception is transformed and rethrown as a C++ exception; col. 13, lines 8-10). Also note section 3 of this office action.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Bak because it would improve the performance the system of Kannan by being able to handle the exception in multiple programming languages.

As to claim 3, Kannan teaches determining a corresponding exception handler to which the exception is to be dispatched (The operating system ... the chain 114 of exception handlers; col. 7, lines 5-8 and The operating system includes ... by the OS itself; col. 4, lines 24-30), dispatching the exception to the corresponding exception handler when the corresponding exception handler exists (When exception handler 115 is trigger; col. 7, lines 12-14), and dispatching the exception to a top level dispatcher when no corresponding exception handler exists (Certain types of application errors ... fatal exception; col. 1, line 47 - col. 2, line 11).

As to claim 4, Kannan teaches dispatching the trapped exception to a trapped exception handler (When exception handler 115 is trigger; col. 7, lines 12-14 and The exception handler 115 ... terminate the application; col. 4, lines 44-47).

As to claim 7, Kannan does not explicitly teach the translating step translates the trapped exception into an exception which is able to be resolved by a C++ based exception handler, and determining if there is a C++ based exception handler which is capable of resolving the translated exception. Bak teaches the translating step translates the trapped exception into an exception which is able to be resolved by a C++ based exception handler (Transform the C++ exception to a Java exception before the exception is passed on; Col. 11, lines 50-55), and determining if there is a C++ based exception handler which is capable of resolving the translated exception (When an exception is generated ... for that exception; Col. 11, lines 25-29). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Bak because it would improve the performance the system of Kannan by being able to handle the exception in multiple programming languages. Note discussion of claim 1 for the application and exception being C++ in nature.

As to claim 13, see rejection of claim 1 above. Kannan further teaches the application being executed under an operating system having one or more low level exception handlers and a top level exception handler (Loaded into and executing ...processor of a computer; Col. 4, lines 4-12 and The operating system ... itself; col. 4, lines 23-30). Note discussion of claim 1 for the application and the exception being C++ in nature.

As to claim 14, see rejection of claim 7 above.

As to claim 21, it corresponds to the method claim of claim 1 except it is a computer system claim.

As to claim 23, Kannan teaches the exception trapper is provided in place of atop level exception handler which terminates the application (The exception handler 115 is inserted in the exception handler chain 114 ahead of all operating system provided exception handlers; col. 6, lines 37-42).

As to claim 24, Kannan teaches an application system for recovering an application from a runtime fault caused in a thread (A method ... a fatal exception; abstract), the application running under an operating system having an exception dispatcher (inherent from the operating system ... of exception handlers; col. 7, lines 5-8), one or more low level exception handlers and a top level exception handler which terminates the application (The operating system ... itself; col. 4, lines 23-30 and Certain types of application errors ... fatal exception; col. 1, line 47 - col. 2, line 11), an exception trapper placed between the exception dispatcher and the top level exception handler for trapping an exception before the exception reaches the top level exception handler (The exception handler 115 is inserted in the exception handler chain 114 ahead of all operating system provided exception handlers; col. 6, lines 37-42 and The exception handler 115 ... terminate the application; col. 4, lines 44-47), the exception being caused due to a runtime fault in a thread executing an application (The current instruction 302 ... a general protection fault; col. 6, line 66 - col. 7, line 2), and a trapped exception handler for handling the trapped exception (The exception handler 115 ... terminate the application; col. 4, lines 44-47 and the crash guard process 107; col. 4, lines 38-64).

However, Kannan does not teach an exception translator for translating the trapped exception into an exception that the application is capable of handling. Bak teaches translating the trapped exception into an exception that the application is capable

of handling (Col. 11, lines 50-55). Note discussion of claim 1 for the application and the exception being C++ in nature.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Bak because it would improve the performance the system of Kannan by being able to handle the exception in multiple programming languages.

As to claim 26, see rejection of claim 3 above.

As to claims 33 and 34, they correspond to the method claim of claim 1 except they are computer readable memory element of computer electronic signals claims, respectively.

5. Claims 2, 5, 6, 8, 15, 16, 22, 25, 27 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kannan et al. (U.S. 5,815,702) in view of Bak et al. (U.S. 6,415,381 BI) further in view of Anschuetz et al. (U.S. 5,305,455).

As to claim 2, Kannan does not teach terminating the thread that caused the exception. Anschuetz teaches terminating the thread that caused the exception (Col. 5, lines 9-16). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Anschuetz because it provides a method to handle the exception on a per thread basis.

As to claim 5, Kannan does not teach terminating the thread when the trapped exception handler is not capable of resolving the trapped exception. Anschuetz teaches terminating the thread that caused the exception (col. 5, lines 9-16). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine

the teaching of Kannan and Anschuetz because it provides a method to handle the exception on a per thread basis.

As to claim 6, Kannan does not teach the continuing step allows continuing execution of the application after the thread is terminated. Kannan teaches the application is continue execution after the application generates a fatal exception which always causes termination to the application (abstract). Anschuetz teaches terminating the thread that caused the exception (col. 5, lines 9-16). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Anschuetz because it provides a method to handle the exception on a per thread basis, and the application has change to continue execution.

As to claim 8, Kannan does not teach terminating the thread that caused the exception when there is no C++ based exception which is capable of resolving the translated exception. Kannan teaches the application is terminated when there is no C++ based exception which is capable of resolving the exception (col. 1, line 47 - col. 2, line 11). Anschuetz teaches terminating the thread that caused the exception (col. 5, lines 9-16). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Anschuetz because it provides a method to handle the exception on a per thread basis, and the application has a chance to continue execution. Note discussion of claim 1 for the application and exception being C++ in nature.

As to claim 15, see rejection of claim 8 above.

As to claim 16, see rejection of claim 2 above.

As to claim 22, see rejection of claim 2 above.

As to claim 25, see rejection of claim 2 above.

As to claim 27, see rejection of claim 2 above.

As to claim 28, Kannan teaches the trapped exception handler further comprises a state restorer for restoring the state that the application was in before the fault occurred to continued the execution of the application (col. 7, lines 3-5).

6. Claims 9-12, 17-20 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kannan et al. (U.S. 5,815,702) in view of Bak et al. (U.S. 6,415,381 B 1) and Anschuetz et al. (U.S. 5,305,455) further in view of LeVine et al. (U.S. 6,591,379 B 1).

As to claim 9, Kannan does not teach logging state information representing the state that the application was in before occurrence of the exception caused the termination of the thread. LeVine teaches logging state information representing the state that the application was in before occurrence of the exception caused the termination of the thread (col. 7, lines 2-7 and col. 8, lines 41-46 and Fig. 5). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and LeVine because LeVine's logging state information would improve the reliability of Kannan's system by saving all necessary information for later recovery.

As to claim 10, Kannan does not teach forwarding the logged information to a remote database over a computer network. LeVine teaches forwarding the logged information to a remote database over a computer network (col. 7, lines 5-8).

As to claim 11, Kannan teaches receiving a recommendation from the remote database (col. 6, lines 15-18), and informing the recommendation to the user (col. 7, lines 34-44).

As to claim 12, Kannan does not teach forwarding a bug report to a bug report center over a computer network. LeVine teaches forwarding a bug report to a bug report center over a computer network (col. 8, lines 1-8).

As to claim 17, see rejection of claim 9 above.

As to claim 18, see rejection of claim 10 above.

As to claim 19, see rejection of claim 11 above.

As to claim 20, see rejection of claim 12 above.

As to claim 29, see rejection of claim 9 above.

7. Claims 30-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kannan et al. (U.S. 5,815,702) in view of Bak et al. (U.S. 6,415,381 B1), Anschuetz et al. (U.S. 5,305,455) and LeVine et al. (U.S. 6,591,379 B1) further in view of Lillevold (U.S. 6,230,284 B1).

As to claim 30, Kannan does not teach a query generator for generating a query including the state information to query a recommendation from a remote database over a computer network. Lillevold teaches the crash handler program determines the state of the computer, sends the information to the server, and the server, based on this information, send revision code to the computer to avoid the crash next time (col. 3, lines 21-27). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Kannan and Lillevold because it would

improve the performance of Kannan system by obtaining correct help from server when it is not available locally.

As to claim 31, see rejection of claim 11 above.

As to claim 32, see rejection of claim 12 above.

8. Applicant's arguments filed 12/29/2004 have been considered but are moot in view of the new ground(s) of rejection. Applicant amended claims have added limitations not previously recited, thus, requiring a new grounds of rejection.

The newly amended C++ based application and C++ exceptions, this is met by Bak who teaches translating a trapped exception into a C++ exception (Java exception is transformed and rethrown as a C++ exception; col. 13, lines 8-10), as discussed in the rejection of claim 1. Also note section 3 of this office action for issues concerning the first paragraph of 35 U.S.C. 112.

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Diem K Cao whose telephone number is (571) 272 3760.

The examiner can normally be reached on Monday - Thursday, 9:00AM – 5:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272 3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR.) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any response to this action should be mailed to: Commissioner for Patents PO Box 1450 Alexandria, VA 22313-1450.

January 13, 2005



**SUE LAO
PRIMARY EXAMINER**